RECEIVED
CENTRAL FAX CENTER

MAR 2 3 2006

PATTERSON & SHERIDAN, LLP
ATTORNEYS AT LAW

3040 Post Oak Blvd, Suite 1500
Houston, TX 77056-6582
TEL 713.623.4844
FAX 713.623.4846

FACSIMILE COVER SHEET

DATE: March 23, 2006

FILE NO: ROC920010101US1 (IBMK10101)

TO: MAIL STOP APPEAL BRIEF - PATENTS

Examiner Paul H. Nguyen-Ba

FAX NO: 1-571-273-8300

FROM: Gero G. McClellan/Jon Stewart

PAGE(S) with cover: 22

RE:

TITLE: METHOD AND SYSTEM FOR ASCERTAINING CODE SETS ASSOCIATED WITH REQUESTS AND RESPONSES IN MULTI-LINGUAL DISTRIBUTED ENVIRONMENTS

U.S. SERIAL NO.: 09/904,734
FILING DATE: July 13, 2001
INVENTOR(S): Banerjee et al.
EXAMINER: Paul H. Nguyen-Ba
GROUP ART UNIT: 2176
CONFIRMATION NO.: 3372

Attached are the following document(s) for the above-referenced application:

Appeal Brief.

451078_1.DOC

RECEIVED
CENTRAL FAX CENTER
MAR 2 3 2006

PATENT
Atty. Dkt. No. ROC920010101US1
PS Ref. No.: IBMK10101

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:
Banerjee et al.

Serial No.: 09/904,734

Confirmation No.: 3372

Filed: July 13, 2001

For: METHOD AND SYSTEM FOR ASCERTAINING CODE SETS ASSOCIATED WITH REQUESTS AND RESPONSES IN MULTI-LINGUAL DISTRIBUTED ENVIRONMENTS

Group Art Unit: 2176

Examiner: Paul H. Nguyen-Ba

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Paul H. Nguyen-Ba, on the date shown below:

March 23, 2006
Date

Jon K. Stewart

APPEAL BRIEF

Dear Sir:

Applicants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2176 dated September 22, 2005, finally rejecting claims 1 – 27. The final rejection of claims 1 – 27 is appealed. This Appeal Brief is believed to be timely since facsimile transmitted by the due date of March 23, 2006, as set by mailing a Notice of Appeal on January 23, 2005. Please charge the fee of $500.00 for filing this brief to Deposit Account No. 09-0465/ROC920010101US1.

PATENT
Atty. Dkt. No. ROC920010101US1
PS Ref. No.: IBMK10101

TABLE OF CONTENTS

Real Party in Interest

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-27 are pending in the application. Claims 1-27 were originally presented in the application. Claims 1-27 stand finally rejected as discussed below. The final rejections of claims 1-27 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments have been entered by the Examiner. No amendments to the claims were proposed after the final rejection.

Summary of Claimed Subject Matter

Claimed embodiments of the invention provide a method, server computer system, and computer program product stored on a computer readable medium for determining a character set associated with a client request or server response (e.g., an HTTP request or response message). See, e.g., *Application*, ¶ 1, 13, 25, Abstract.

One embodiment (*see, e.g.*, claim 1) provides a method of determining an appropriate character set for use in client-server communications. *See Application*, ¶ 14, 25, 39, Figs. 4, 5. The method generally includes at least one of (a) selecting a character set for a client request made by client to a server using a network communication protocol (*see, e.g., Application*, ¶ 39, Fig. 4), and (b) selecting a response character set for a response from the server to the client (*see, e.g., Application*, ¶ 45, Fig. 5.).

In the case of selecting a character set for a client request, the claimed embodiment includes determining whether the client request includes, as part of the network communication protocol, a request character set designation. *See e.g., Application*, ¶ 34, 35, 39, 40 Fig. 4, elements 402, 406.) As recited by claim 1, the determination is made whether the client request itself includes a request character set designation based on the network communication protocol.

If the client request does not include the request character set designation, then this claimed embodiment includes (i) retrieving locale information contained in the client request (*See, e.g., Application*, ¶ 35, 40 Fig. 4, element 408), and (ii) associating the locale information with the request character set designation using mapping data located on the server (*See, e.g., Application*, ¶ 35, 41, 44, Fig. 4 element 410).

In the case of selecting a response character set for a response from the server to the client, the claimed embodiment includes determining whether the server response includes a response character set designation. *See e.g., Application*, ¶, 35, 45, Fig. 5, elements 502, 506.

If the server response does not include the response character set designation, then this claimed embodiment includes (i) retrieving locale information contained in the server response (*See, e.g., Application*, ¶ 45, 46, Fig. 5 elements 508 and 512), and (ii) associating the locale information contained in the server response with the response character set designation using the mapping data (*See, e.g., Application*, ¶ 46, 47, Fig. 5 elements 520 and 522).

Another embodiment (*see e.g.*, claim 12) includes a server computer system connected to at least one client computer, the server computer system comprising a memory containing a code-set program and at least one processor (*See, e.g., Application*, ¶ 15, 25, 26 30-32, Fig. 1). In this claimed embodiment, when executed, the code-set program is configured to determine if a request header composed according to a network communications protocol received with a client request from the at least one client computer designates a character set. *See e.g., Application*, ¶ 34, 35, 39, 40 Fig. 4, elements 402 and 406.) As recited by claim 12, the determination is made whether the request header, which itself is composed according to a network communications protocol, designates a character set.

If the request header does not designate the character set, then, according to this claimed embodiment, the code set program is configured to (i) retrieve locale information from the client request; and (*See, e.g., Application*, ¶ 35, 40 Fig. 4, element 408), and (ii) associate the locale information with a character set (*See, e.g., Application*, ¶ 35, 41, 44, Fig. 4 element 410).

Another embodiment (*see e.g.*, claim 16) includes a computer readable medium containing at least a code-set program which, when executed by a server computer, performs operations. (*See, e.g., Application*, ¶ 15, 25, 26). As claimed, the operations perform the method steps generally recited by claim 1.

Grounds of Rejection to be Reviewed on Appeal

1. Claims 1, 3-5, 7-9, 12-14, 16, 18-20 and 22-24 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz et al.*, U.S. Pat. No. 6,496,793 (hereinafter *Veditz*) in view of *Watanabe et al.*, U.S. Pat. No. 6,185,729 (hereinafter *Watanabe*).

2. Claims 2, 6, 10, 11, 17, 21, 26 and 27 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz* in view of *Watanabe*, and further in view of *Horn*, U.S. Patent Pub. No. 2002/0156688.

3. Claims 15 and 25 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz* in view of *Kan*, U.S. Patent Pub. No. 2003/0088544.

ARGUMENTS

Obviousness of Claims 1-27

The Examiner bears the initial burden of establishing a *prima facie* case of obviousness. *See* MPEP § 2142. To establish a *prima facie* case of obviousness three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one ordinary skill in the art to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* MPEP § 2143. Applicants submit that the final rejection in this case fails to establish at least the first and third criteria.

The References

Veditz discloses a non-networked system configured to "intelligently process data objects created or modified under one language driver with those created or modified by a different language driver." *Veditz*, Abstract. As disclosed, the "data objects" are elements of a database, e.g., a database file, or data elements contained in database tables, rows, and columns. The system disclosed by *Veditz*:

> continually checks and maintains correct language configuration. A descriptor or Language Driver Identifier (LDID) (e.g., in the form of a system-comparable unit) is employed for storing in desired location(s) of a data object information specifying the langue driver that was in use when the data object was created or modified.

Veditz, 3:23-28. This process of maintaining a correct language configuration enables the system of *Veditz* to determine when the system is inappropriately configured for a data object about to be processed. *Veditz*, 7:45-50. Each data object in the system may be embedded with an LDID, and the system itself maintains an "active LDID" (i.e., the LDID presently being used by the system). *Veditz*, 14:56-62. The active LDID, in turn, is written to any data object that the system "touches" (i.e., creates or modifies).

Veditz, 14:63-64. In the event of a mismatch between the "active" LDID and the LDID of a "data object," corrective actions may be taken.

As disclosed in *Veditz*, each "data object 201 is preferably constructed so that it embeds or stores a Language Deriver Identifier [LDID] 215 for indicating the language support under which the file was created (or last modified)." *Veditz* 7, 29-33. By embedding an "LDID" in each "data object," the system disclosed by *Veditz* allows "data objects" to be appropriately configured, based on an "active LDID" and the "LDID" of a given data object. For example, *Veditz*, Figure 3 illustrates an operation of opening a database file. Specifically, *Veditz* provides: "After receiving a request to open a file in step 301, the method proceeds to step 302 to determine whether language driver checking is enabled." *Veditz*, 16:49-51. If "language driver checking" is enabled, "then at step 303, the language driver identifier (LDID) in the data file is read." *Veditz*, 16:55-57. Importantly, *Veditz* discloses that the LDID is read from the data object. In fact, *Veditz* goes on to describe storing the LDID in different sections of the data file:

> In a preferred embodiment, the identifier will be stored in the data file at a position where it may be conveniently accessed upon first reading the file. The identifier may be stored, for instance, within a header of the data file. Those skilled in the art will appreciate, however, that the identifier may be positioned at a different location or locations within the data file. In the instance of a data file comprising a plurality of data regions (either logically or physically discrete), the language driver identifier may be stored within any organizable unit of data where language configuration is important, including within selected records or fields (individually or by group) and the like. Alternatively, the identifier may be stored in a footer to the file but in such a case should preferably be read before processing other information contained within that file is undertaken.

Veditz, 16:57-67 – 17:1-5. As this passage makes clear, *Veditz* discloses storing the "LDID" value with a "data object" (e.g., database file).

Watanabe discloses a "development suite for developing and testing internationalized software includes, in addition to an ASCII English locale, a multibyte English locale. The presence of a multibyte English locale permits early discovery and correction of errors by English speaking developers which would otherwise only be found during localization of the software for a country where a multibyte representation was required." *Watanabe*, Abstract. As disclosed in *Watanabe*, an internationalization

process is used to create computer software applications that may be localized for use in different parts of the world, with different languages, alphabets, and other conventions.

> To enable this, an internationalization model includes three parts. Namely, a language independent program, message catalogs and language tables. FIG. 1 illustrates a model of internationalized software. A language independent program 100 achieves language independence by programmatic calls to a message catalog 110 and to language table 120. Rather than hard-coding messages such as prompts and error messages within the program itself, such messages are stored in external message catalogs with a different version of those catalogs for each supported language. Language tables contain all language-specific processing information and conventions unique to a particular locale, such as how characters are sorted and how output (such as numbers, times and dates) is formatted. At run time, generally in a development environment the program selects or "binds" a specific language table according to settings controlled by the user, the application developer, or system administrator. Thus, the same basic program 100 can be executed in different language "locales" by simply binding the appropriate message catalog and language table to the program at run time. The term "locale" will be utilized to refer to the language table component of an internationalized application.

Watanabe, 2:27-49. Importantly, *Watanabe* limits the term "locale" to refer to "the language table component of an internationalized application." Id. As described in *Watanabe* problems arise in developing and testing internationalized applications, because English language character sets (at least at the time of *Watanabe*) are typically represented using a single byte, where many Asian languages are represented two (or more) bytes. Thus, applications being developed and tested using a single byte character set for were difficult to test and debug when localized to a mutli-byte character set. See *Watanabe*, 4:23-28. *Watanabe* addresses this difficulty by providing "a multi-byte locale for a single byte language which would act for testing purposes just like a multi-byte locale for a multibyte language but in which the content was in the single byte language." *Watanabe*, 4:48-48.

In a section titled "Summary of Invention", *Watanabe* provides that the: "invention is also directed to a computer system for developing and testing an internationalized computer program written in a single byte language including a network, [and] one or more computers connected to the network...." *Watanabe* 5:35-38. This passage

paraphrases *Watanabe*, claim 4. Claims 5-7 also include a limitation reciting a "network." However, other than eight uses of the word "network" in these claims, Watanabe fails to describe any other aspects data communications, communication protocols or the functional characteristics thereof.

Argument

The Examiner asserts that *Veditz*, in view of *Watanabe*, renders claims 1, 3-5, 7-9, 12-14, 16, 18-20 and 22-24 obvious. Respectfully, Applicants disagree.

The Examiner asserts that *Veditz* discloses the limitation recited by claims 1 and 16 of:

> determining whether the client request includes, as part of the network communication protocol, a request character set designation;

Regarding the determining element, the Examiner asserts that "Veditz Fig 3A-303 → checks LDID in data file (i.e. stored in header file); Fig 2C → file header". *Advisory Action*, continuation sheet. *See Also Final Office Action*, p. 3. The "LDID" value, however, is not "retrieved from the client request" as recited by claims 1 and 16. Plainly, the "LDID" value in the material cited by the Examiner is retrieved from the data object and not from the request. As set forth above, *Veditz* is clear about the LDID value being part of the data object. Applicants submit therefore, that *Veditz* fails to disclose "determining whether the client request includes ... a request character set designation," as recited by claims 1 and 16.

Moreover, claims 1 and 16 further recite the limitation of:

> if the client request does not include the request character set designation:
> (i) retrieving locale information contained in the client request; and ...

Regarding this limitation, the Examiner asserts that *Veditz* "Fig. 3B → compares LDID of data file to Active LDID; see also col. 3, lines 29-21" *Advisory Action*, continuation sheet; *See Also Final Office Action*, p. 3. However, nothing in this material from *Veditz* describes anything being retrieved, let alone, locale information being retrieved from a client request. Instead, this material describes an "active LDID" being compared to the LDID of a "data file". Nothing about a client request (as recited by claims 1 and 16) is

described, or would even be germane, to the comparison of an "active LDID" and the "LDID of a "data file" (as disclosed in *Veditz*).

Finally, claim 1 and 16 are directed to a method ... "for use in client-server communications that includes selecting a character set for a client request ... using a network communication protocol." The method includes "determining ... as part of the network communication protocol" whether the client request includes a "request character set designation." The Examiner recognizes that *Veditz* fails to disclose anything related to client server communications and network communication protocols.

However, the Examiner turns to *Watanabe* and asserts: "*Watanabe* teaches a method and system for developing and testing internationalized software including a multibyte English locale directed to a network communication protocol for the purpose of transferring locale information over computer networks." *Advisory Action*, continuation sheet; *See Also Final Office Action*, p. 4. Respectfully, even a broad reading of *Watanabe* simply does not support such a teaching. *Watanabe* is not directed to a network communications protocol; rather, as described above, *Watanabe* is directed to a "computer system for developing and testing an internationalized computer program written in a single byte language." *Watanabe* happens to include "a network" as an element of some claims. *See, e.g., Watanabe*, claims 4-7. However, *Watanabe* fails to disclose any aspects of a "communication protocol" or the operational or functional characteristics thereof. In fact, other then the appearance of the word "network" in the claims 4-7 (and in a summary of these claims) *Watanabe* is silent about data communications. Applicants assert, therefore, that *Watanabe* fails to disclose any details of client server communications, communications protocols, client requests or server responses as recited by claims 1 and 16.

Accordingly, for all the reasons set forth above, Applicants submit that claims 1 and 16 are allowable over *Veditz* in view of *Watanabe* and respectfully request allowance of the same. In addition, the Examiner cites the same passages discussed above to support the rejection of claim 12. Applicants respectfully submit, therefore, that *Veditz* in view of *Watanabe* fails to disclose the limitation recited by claim 12 of a computer program configured to "determine if a request header composed according to

a network communications protocol received with a client request from the at least one client computer designates a character set; and if the request header does not designate the character set: (i) retrieve locale information from the client request..." Accordingly, claim 12 is believed to be allowable, and allowance of this claim is respectfully requested.

Claims 3-5, 7-9, 13-14, 18-20 and 22-24 each depends from one of independent claims 1, 12, and 16 and, therefore, are believed to be allowable.

Claims 2, 6, 10, 11, 17, 21, 26 and 27 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz* in view of *Watanabe*, and further in view of *Horn*. Each of these claims depends from one of independent claims 1, 12, and 16. Applicants respectfully submit, for all the reasons given above, that claims 1, 12, and 16 are allowable, and therefore, that claims 2, 6, 10, 11, 17, 21, 26 and 27 are also allowable.

Claims 15 and 25 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz* in view of *Kan*, U.S. Patent Pub. No. 2003/0088544. Each of these claims depends from one of independent claims 1 and 16. Applicants respectfully submit, for all the reasons given above, that claims 1 and 16 are allowable, and therefore, that claims 15 and 25 are also allowable. Additionally, Applicants note that the rejection formally refers only to *Veditz* and *Kan*, where the substance of the rejection provides "*Veditz* teaches the system with respect to independent claim 12 as discussed above...." *See Final office Action*, p. 11. The rejection of claim 12, however, is based on *Veditz*, in view of *Watanabe*. Applicants believe the rejection was intended to rely on *Veditz* in view of *Watanabe*, and further in view of *Kan*.

CONCLUSION

The Examiner errs in finding that:

- Claims 1, 3-5, 7-9, 12-14, 16, 18-20 and 22-24 are unpatentable over *Veditz* in view of *Watanabe*.
- Claims 2, 6, 10, 11, 17, 21, 26 and 27 are unpatentable over *Veditz* in view of *Watanabe*, and further in view of *Horn*.
- Claims 15 and 25 are unpatentable over *Veditz* in view of *Watanabe*, in further view of *Kan*.

Withdrawal of the rejection and allowance of all claims is respectfully requested.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Gero G. McClellan, Reg. No. 44,227/

Gero G. McClellan
Registration No. 44,227
Patterson & Sheridan, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Appellant(s)

CLAIMS APPENDIX

(Previously Presented) A method of determining an appropriate character set for use in client-server communications, comprising at least one of:

(a) selecting a character set for a client request made by client to a server using a network communication protocol, the selecting comprising:

determining whether the client request includes, as part of the network communication protocol, a request character set designation; and

if the client request does not include the request character set designation:

(i) retrieving locale information contained in the client request; and

(ii) associating the locale information with the request character set designation using mapping data located on the server; and

(b) selecting a response character set for a response from the server to the client, the selecting comprising:

determining whether the server response includes a response character set designation; and

if the server response does not include the response character set designation:

(i) retrieving locale information contained in the server response; and

(ii) associating the locale information contained in the server response with the response character set designation using the mapping data.

2. (Previously Presented) The method of claim 1, wherein the network communications protocol used to make the client request and the server response comprises the hypertext transfer protocol (HTTP).

3. (Original) The method of claim 1, wherein associating comprises accessing a character set lookup table that maps the locale information to the request character set designation and response request character set designation, respectively.

4. (Original) The method of claim 1, further comprising associating the request character set designation with a code-set converter designation by accessing a converter lookup table which maps the code-set converter designation with the request character set designation.

5. (Original) The method of claim 1, wherein the locale information contains a cultural language preference identifier.

6. (Original) The method of claim 1, wherein the character set designations contain an IANA character set parameter.

7. (Original) The method of claim 1, further comprising associating the request character set designation with a code-set converter designation.

8. (Original) The method of claim 7, wherein the code-set converter designation is contained in a lookup table and is mapped with response character set designation.

9. (Original) The method of claim 7, wherein the code-set converter designation is indicative of user specific implementations of character sets.

10. (Original) The method of claim 1, further comprising converting the client request into Unicode characters.

11. (Original) The method of claim 10, further comprising converting the response from Unicode characters to the character set associated with the locale information.

12. (Previously Presented) A server computer system connected to at least one client computer, the server computer system comprising a memory containing a code-set program and at least one processor, wherein the processor, when executing the code-set program, is configured to:

determine if a request header composed according to a network communications protocol received with a client request from the at least one client computer designates a character set; and

if the request header does not designate the character set:

(i) retrieve locale information from the client request; and

(ii) associate the locale information with a character set.

13. (Original) The system of claim 12, wherein the processor is further configured to associate the character set with a code-set converter.

14. (Original) The system of claim 12, wherein the locale information contains a language identifier.

15. (Original) The system of claim 12, wherein the code-set converter is a JVM code-set converter.

16. (Previously Presented) A computer readable medium containing at least a code-set program which, when executed by a server computer, performs operations comprising at least one of:

(a) selecting a character set for a client request made by client computer to a server computer using a network communication protocol, the selecting comprising:

determining whether the client request includes, as part of the network communication protocol, a request character set designation, and

if the client request does not include the request character set designation:

(i) retrieving locale information contained in the client request; and

(ii) associating the locale information with the request character set designation using mapping data located on the server; and

(b) selecting a response character set for a server response from the server to the client, the selecting comprising:

determining whether the server response includes a response character set designation; and

if the server response does not include the response character set designation:

(i) retrieving locale information contained in the server response; and

(ii) associating the locale information contained in the server response with the response character set designation using the mapping data.

17. (Previously Presented) The method of claim 1, wherein the network communications protocol used to make the client request and the server response comprises the hypertext transfer protocol (HTTP).

18. (Original) The computer readable medium of claim 16, wherein associating comprises accessing a character set lookup table that maps the locale information to the request character set designation and response request character set designation, respectively.

19. (Original) The computer readable medium of claim 16, further comprising associating the request character set designation with a code-set converter designation by accessing a converter lookup table which maps the code-set converter designation with the request character set designation.

20. (Original) The computer readable medium of claim 16, wherein the locale information contains a cultural language preference identifier.

21. (Original) The computer readable medium of claim 16, wherein the character set designations contain an IANA character set parameter.

22. (Original) The computer readable medium of claim 16, further comprising associating the request character set designation with a code-set converter designation.

23. (Original) The computer readable medium of claim 22, wherein the code-set converter designation is contained in a lookup table and is mapped with response character set designation.

24. (Original) The computer readable medium of claim 22, wherein the code-set converter designation is indicative of user specific implementations of character sets.

25. (Original) The computer readable medium of claim 24, wherein the code-set converter designation is contained in a Java Virtual Machine (JVM) code-set converter.

26. (Original) The computer readable medium of claim 16, further comprising converting the client request into Unicode characters.

27. (Original) The computer readable medium of claim 26, further comprising converting the response from Unicode characters to the character set associated with the locale information.